AD-A128 881 SOFTWARE TECHNOLOGY FOR ADAPTABLE RELIABLE SYSTEMS (STARS) FUNCTIONAL TASK AREA STRATEGY FOR MEASUREMENT (U) OFFICE OF THE DEPUTY UNDER SECRETARY OF DEFENSE RESEARCH AND E .. 30 MAR 83 1/1

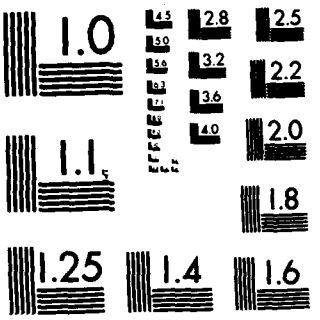
UNCLASSIFIED F/G 9/2 NL

END

DATE FILMED

6-83

DTIC



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. A128881
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Software Technology For Adaptable Reliable Systems (STARS) Functional Task Area Strategy For MEASUREMENT	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) The DoD Joint Service Task Force on the Software Initiative (STARS)	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy Under Secretary of Defense Research & Advanced Technology Washington, D.C. 20301	12. REPORT DATE March 30, 1983
	13. NUMBER OF PAGES
MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unclassified

SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

STARS, measurement, reliable, adaptable, Ada, support environments, hardware, software, software engineering institute, military systems, mission critical, embedded computers, automated tools, system interface standards, software initiative, functional task areas, applications, embedded instrumentation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document identifies the scope, sub-objectives and strategies designed to provide the conceptual approach for accomplishment of the STARS Program objectives in the measurement functional task area. Its main objective is to help guide the implementation planning process.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

FUNCTIONAL TASK AREA STRATEGY FOR MEASUREMENT

Department of Defense

30 March 1983

FOREWORD

This strategy document is one of eight functional task area strategies produced by the STARS Joint Task Force. All of the documents produced by the Task Force, including the general STARS Program Strategy document, are listed in the STARS Joint Task Force Report.

This document identifies the scope, sub-objectives and strategies designed to provide the conceptual approach for accomplishment of the STARS Program objectives in the measurement functional task area. It identifies and describes the high-level activities, products and capabilities. In order to provide full understanding, background and rationale material is sometimes covered that is also in STARS Program Strategy.

These functional task area strategy documents do not attempt to delineate the detailed plans, costs and procedures for bringing the proposed products and capabilities into being and do not identify the form of the particular projects that will undertake the work nor the organizations in which the work will be accomplished. Instead, these strategies are intended to guide the process of such implementation planning and accomplishment.

Indeed, because of the high degree of linkage among the functional task areas, implementation plans and acquisitions may well combine related capabilities and products across areas. Individual projects may tackle only part of one subtask from a functional area or several subtasks from several functional areas.

Thus, this functional task area strategy describes broad, achievable requirements for accomplishing the relevant STARS objectives. Its main purpose is to help guide the implementation planning process.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

1.0 PLAN OVERVIEW

1.1 Scope of the Measurement Task Area

There is a grave need for measurement in all phases of software development, from requirements definition through operations and maintenance. For example, studies have shown that testing and maintenance constitute at least 75% of the effort involved in software production, and that over 60% of all errors discovered are made prior to construction. Unfortunately, there is little technology in use which can help identify problems up front, when they are least expensive to correct. The measurement task area recognizes such needs by sponsoring measurement-related activities. Through measurement, one can answer important questions related to the product (e.g. How good is the product at this point? Should I accept it? Is it too complex?). In addition, measurement in its refined stages can permit the prediction of costs, end-product quality and maintainability, and resources required. Finally, measurement would permit the determination, evaluation, and/or selection of approaches and technologies which could be most effective given the characteristics of the project and a list of alternative methodologies to employ for developing the software.

In the ensuing discussion, the terms measurement, metric or measure, and model have different connotations. Measurement connotes the act of measuring the degree to which an entity or process exhibits an attribute or factor of interest. The term metric or measure defines the criterion which is measured and which relates to the desired attribute or factor of interest. A model is an analytical equation which explains the relationship between the criteria measured and the desired attribute or factor of interest. For example, the measurement of personnel resources required to develop a project uses the lines of code metric. The model combines the lines of code metric with other metrics and past history data to develop the

relationship between lines of code, the other metrics, and the amount of personnel resources required. This model can then be used to predict personnel resources given estimates of the lines of code metric and estimates of the other metrics for projects of a similar nature.

The measurement task area is concerned with activities to develop models and metrics, creating and maintaining software data collection and analysis activities, supporting the use of the metrics and models during the total life cycle, and providing customized measurement support for the Software Technology for Adaptable and Reliable Systems (STARS) program. These four activities run concurrently throughout the duration of the measurement task. They define the scope of the measurement task area.

The measurement task area should sponsor activities to develop measures of the software product, development and support processes, and resources. Measures of the software product include measures related to product size, product quality (e.g., reliability, testability), performance, and cost. Measures of the development process include measures of effort, time, schedule, errors, changes made, and development methods employed. Measures of the resources include measures related to the of use of the personnel and other computer resources. The activities sponsored should include model and metric definition, validation, and calibration. The measurement task should also sponsor the development of specialized instrumentation, data collection, and analyses required to support these metric and model development activities.

The measurement task area should support data collection and analysis activities required for developing, refining and maintaining a set of baselines. The baselines should provide lifecycle information on the cost, quality, and resources for a representative sample of software projects. This information would be useful to software

and systems managers for estimating cost and resources required for achieving acceptable software quality on new projects and for assessing STARS progress.

The measurement task area should support the development of instrumentation tools required for collecting the data required to drive the models and metrics. The instrumentation tools would implement both manual and automated data collection during the software life cycle. The automated tools developed would include both stand-alone and embedded instrumentation. The embedded instrumentation would be associated with the support environment supported by the Software Initiative. A stand-alone micro processor based data collection and analysis tool would provide the capability to instrument other support environments at minimal cost. Since instrumenting a variety of support environments requires an exorbitant amount of funding, the stand-alone unit should provide a cost-effective solution to serving the needs of more than one community.

The measurement task area should support the use of models and metrics during the acquisition, development and support cycles by disseminating model and metric definition and analysis descriptions, guides on data collection, tool usage, and model and metric use. In addition, the measurement task area should support the development of training for model and metric use, data collection and analysis, and tool usage; and clinics for fine-tuning the tools, models, and metrics to a particular environment. This support would facilitate the insertion of models and metrics as an integral part of the acquisition, development and support processes.

The measurement task area should support the measurement requirements of the overall STARS Program and specific STARS task areas on a demand basis. The support to the overall program would be concerned with determining potential and/or actual return on the DoD investment. The customized measurement support to the task areas

should include recommending experimental paradigms and providing assistance in conducting experiments, data collection, and analytical techniques.

1.2 Strategy for the Measurement Task Area

It is the goal of the measurement task area to insert measurement technology as an integral part of the activities conducted during all phases of the software development, support, and use cycles. The measurement task should accomplish this goal by funding a thorough and exemplary implementation of measurement activities in a few software projects. These projects should demonstrate how measurement enables one to understand and hence improve software engineering during all phases of the software life cycle. It is hoped this demonstration might be picked up by non-STARS Programs so that the base of data and experience in measurement will be broadened so that this task, as well as others, will directly benefit.

The measurement task strategy is partitioned into four major components. These components correspond to developing the models and metrics, collecting and analyzing data for the baselines, supporting the use of models and metrics, and providing measurement support for the whole STARS program. Each of these components can be represented as nodes on the first level of a tree structure as depicted in Figure 1.1. The measurement task plan specifies one additional level of activities for each of these components. Although these components are described independently, some interaction exists between the major components. These interactions are specified as coordination events.

The model and metric development component develops the technology for understanding and providing insight into the software life cycle. Models and metrics of the product, process, and resources should be developed for tracking, prediction, and problem diagnosis.

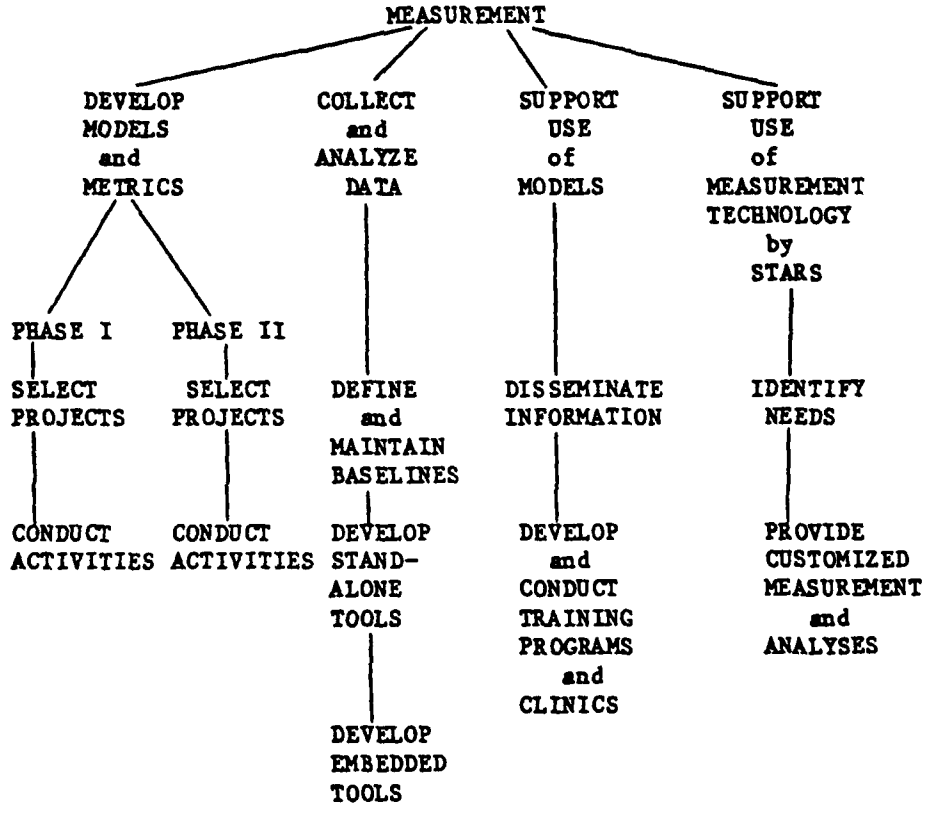
The models and metric development activities should be oriented towards satisfying the needs of several user groups. Measurement needs of the manager, the designer, the programmer, the systems architect, the systems analyst, the user analyst, the customer, the modifier and the acquisition specialist should be addressed. The models and metrics developed will be mostly lang e-independent, however, if language dependency is dictated, Ada wil e the frame of reference.

The data collection and analysis component foc n identifying and resolving issues related to the establishmen maintenance of baselines from which models can be verified, comparisons of resources and progress can be made, projections studied, etc. This component sponsors activities for identifying what data to collect, how to collect the data, who should collect the data, and how to distribute and store the data. The data collection and analysis component should develop instrumentation for extracting characteristic information from software projects. This information would be useful to developers, maintainers, and researchers for prediction, assessment, selection, and control.

The support for use component supports the use of models and metrics during the acquisition, development maintenance, and use cycles by disseminating information related to the baselines. This component also supports the dissemination of reports containing model and metric definition descriptions, and guides on data collection, tool usage, and model and metric use. In addition, the measurement task area should support the development of training aids for model and metric use, data collection and analysis, and tool usage; and clinics for fine-tuning the tools, models, and metrics to a particular environment. This support would facilitate the insertion of models and metrics as an integral part of the acquisition, development, support, and use processes.

The Support for STARS Component provides customized measurement support for STARS. This component focuses the interaction of the measurement task area with other task areas. This focus also permits the transfer of measurement-related technology between task areas.

FIGURE 1.1 THE MEASUREMENT PLAN TREE



2.0 PLAN DETAILS

2.1 Major Subtasks

2.1.1 Sponsor Activities to Develop Models and Metrics

2.1.1.1 Purpose/Goals/Rationale. The model and metric development component develops the technology for understanding and providing insight into software development projects. Models and metrics of the product, process, and resources should be identified, validated, and calibrated for tracking, prediction and problem diagnosis. Measurement should occur both during specific and across several, if not all, life cycle activities. Models and metrics for determining the completion of a phase and for relating events in one phase to events in a later phase should be developed. For example, understanding when and how requirements errors manifest themselves could provide useful insight into software development, and models and metrics of the test process would be invaluable to making assessments of reliability, robustness, cost, etc.

The measurement needs of the manager, systems architect, systems analyst, user analyst, the programmer, the customer, and the acquisition specialist should be addressed during this activity. Goals should be specified to ensure that the model and metric development activities are oriented towards these groups of users. The models and metrics developed should be mostly language-independent, however, if language dependency is dictated, Ada would be the focus.

The model and metric development component can be viewed as an activity graph as shown in Figure 1.2. This graph illuminates the iterative nature of model and metric development and the high degree of interaction required among these development activities. In addition, this graph illustrates the goal-oriented approach to metric development. Although, it may be desirable to collect as much data

as possible, it is not usually cost-effective. Therefore, data collection must occur with respect to the goals of the development task.

While the process of developing the models and metrics is iterative, it is recommended that two fully supported releases of the models and metrics be made, one during FY-86, and the other during FY-88. The second release would take advantage of the experience gained from using the first release, as well as advances in the state-of-the-art. To accomplish this, a two phased approach is recommended.

A major concern is the validity and reliability of the models and metrics developed. This subtask should include the specialized data collection and analysis activities necessary for validating and calibrating those models and metrics developed. Since these activities would cost in the neighborhood of 12% of the cost of the software being analyzed, there is a tradeoff that must initially be made with regard to accuracy vs. dollars available.

2.1.1.2 Input. Information about on-going or future projects which cross several application areas and which are potential candidates for participation in the areas of needed metric development would be required.

2.1.1.3 Description. Select Phase I Projects in Different Application Areas

Establish criteria, and with the help of the service components, identify projects in different application areas (e.g., C3I, avionics, guidance and control, etc.) and solicit participation. Select projects and establish measurement-related goals for each of the projects selected. These goals should be oriented towards satisfying the needs of the different groups of users. This orientation would occur either within a project or across the projects selected. If the sample of candidate projects is not representative, incentives

for project participation should be developed. The development subtask should sponsor as many projects as funds permit, but if necessary, the tradeoff between accuracy desired and cost must be made in this phase.

Conduct Phase I Activities to Develop Models and Metrics

For each project identified in the different applications areas, conduct the activities depicted in Figure 1.2 which contribute to model and metric development. The measures and metrics identified should be reviewed by as wide a group as possible, i.e., DoD, industry, academia.

Select Phase II Projects in Different Application Areas

Review the results of the Phase I efforts, establish criteria for project identification, and solicit participation across application areas. Select projects and establish measurement-related goals for each of the projects selected. These goals should again be oriented towards satisfying the needs of different groups of users. This orientation should occur either within a project or across the projects selected. If the sample of candidate projects is not representative, incentives for project participation should be developed. The development subtask should maximize on the number of application areas and projects participating.

Conduct Phase II Activities to Develop Models and Metrics

For each project identified in the different applications areas, conduct the activities depicted in Figure 1.2 which contribute to model and metric development.

2.1.1.4 Coordination. This subtask requires coordination with the other STARS task areas and on-going service component projects to determine model and metric development needs and to obtain an indication of the important application areas. For example, coordination with the Human Engineering Task Area (e.g., user-oriented measure-

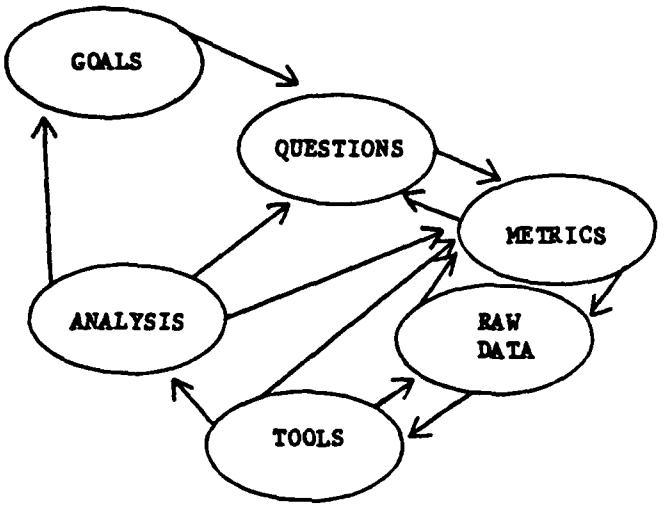
ment) and the Systems Task Area (e.g., test effectiveness, reliability and performance assessment) must occur.

2.1.1.5 Deliverables. Deliverables include a set of advanced models and metrics which have been used on a trial basis for a variety of software development projects and reports describing the results of this activity.

2.1.1.6 Cost Factors. Leverage for this subtask could be achieved by providing incentives for non-STARS projects to develop and use models and metrics on their own, and provide either the raw or processed data to the STARS program office.

2.1.1.7 Benefit. The primary benefit of this subtask is its contribution to improving DoD's ability to measure software, by providing the capability to quantify the software work product and its quality/performance characteristics and by improving product performance and productivity measurements.

FIGURE 1.2 THE MODEL AND METRIC DEVELOPMENT ACTIVITY CHART



2.1.2 Collect and Analyze Data

2.1.2.1 Purpose/Goals/Rationale. The measurement task area should sponsor activities to collect and analyze data required for the computation of the models and metrics for establishing and maintaining baselines for factors such as cost, quality and resources. This component sponsors activities for identifying what data to collect, how to collect the data, who should collect the data, and how the data should be distributed and stored. The data collection and analysis component should define instrumentation for extracting characteristic information from software projects. This information could be used by researchers, developers, and acquisition managers for prediction, assessment, selection, and control.

2.1.2.2 Input. Data from both past and on-going software projects is required to establish and maintain the baseline. Input from the model and metric development subtask, the the baseline definition activity, and the other STARS tasks would be required for specifying the goals of the instrumentation development activity.

2.1.2.3 Description.

Define and Maintain Set of Baselines

The data collection and analysis needed to provide baselines and drive the models and metric analyses should be defined. This data set should be used to characterize the software product; development, and support processes; and development and support resources. The definers of this set should strive for generality to the maximum extent possible and insure that characterization and collection of the data follows the life cycle phases as identified in MIL-STD-SDS, the new tri-service standard under development. This baseline set should be submitted to extensive peer review either by a workshop or a questionnaire. In addition, the baselines must be upgraded on a phased release basis. The Data and Analysis Center for Software

(DACS) could play a role in establishing and maintaining the baselines.

The data required to establish and maintain the baselines should be collected by a detailed questionnaire, environment instrumentation, and acquisition incentives. Data collection activities should be tied to the work breakdown structure of projects. This data collection would occur at a detailed and a gross level. The detailed activity would implement ample data collection across a few projects. The gross activity would implement data collection on a small amount of data on many projects. Integrity in the data collection process should be accomplished by effective management controls, the contractors Q/A staff, or in some cases IV&V.

Data analysis for the baselines should also occur at two levels. Gross analyses must be performed to ensure that the data requirements for maintaining the baseline are being satisfied. In-depth analyses at the project level should be performed to ensure that the project goals are being met and to preserve the anonymity if the data coordination between the data collectors and the data analyzers at the "gross" and project levels must occur, to ensure the validity and integrity of the baselines.

Develop Stand-Alone Instrumentation to Support Collection of Data

Stand-alone instrumentation which implements data collection and applicable analyses should be developed, perhaps on a microprocessor based system. This instrumentation should include facilities for both manual insertion and automated data collection. A possible starting point is the Automated Measurement Tool developed by USAF/RADC and USA/AIRMICS.

Develop Embedded Instrumentation to Support Collection of Data

Instrumentation of an APSE(s) which implements detailed data collection should be developed. This instrumentation should include facil-

ities for both manual and automated data collection and should be embedded within the support systems environments.

2.1.2.4 Coordination. This subtask would require coordination between the Model and Metric development subtask, the Support for Use sub-task, and of course the other STARS task areas. More specifically, coordination with the Acquisition Task Area (e.g., How to get measurement on contract? How to make measurement an integral part of development - Q.A. vs. I.V.V?) should occur. Coordination with the Project Management Task Area (e.g., Q.A. for the data) and, the Support Systems Task Area (e.g., instrumentation tools) should also occur.

2.1.2.5 Deliverables. Deliverables include a standardized description of the data needed to drive the selected set of models and metrics (i.e., a glossary of terms and definitions) for establishing and maintaining the baselines, a set of procedures for collecting data for the baselines, tools for data collection, and the initial baselines themselves.

2.1.2.6 Cost Factors. Leverage for this subtask would be achieved through the previous and succeeding tasks which would simplify the use of the measurements and prove their worth to non-STARS programs. This will result in "free" data on a wider range of applications, thereby increasing the validity of the baselines.

2.1.2.7 Benefit. Benefits include standard baselines useful to all software development projects, and tools for data collection and analysis which will assist acquisition and program managers in performing their own measurements to assess progress, cost, quality, etc., and which would help update and maintain the baselines themselves.

2.1.3 Support Use of Models and Metrics Developed

2.1.3.1 Purpose/Goals/Rationale. In addition to defining, validating, and calibrating metrics, the measurement task plan provides support for the use of metrics throughout software's whole life cycle. This support would facilitate the insertion of models and metrics as an integral part of the acquisition, development, support, and use processes. This subtask should help leverage the funding of the metrics area by allowing non-STARS programs to gain benefits from the measurements, producing as a by-product additional data and analyses which will help refine the models, metrics, and baselines. The DACS or the DoD Software Engineering Institute would be fundamental in providing this support.

2.1.3.2 Input. Input from the other measurement task area subtasks would be required.

2.1.3.3 Description.

Disseminate Information

The information disseminated should include model and metric definition glossaries, guides on data collection, guides on tool availability and usage, and analytical reports describing the baselines, and the models and metrics themselves.

Develop Training Programs and Sponsor Clinics

In addition, the measurement task area should develop training for model and metric use, data collection and tool usage; and clinics for fine-tuning the tools, models, and metrics to a particular environment.

Develop Acquisition Guides

Finally, this task area should develop guides for using the metrics and models on contracts. An incentive structure for rewarding quality and performance will be developed.

2.1.3.4 Coordination. Coordination with the other measurement task area subtasks would be required. Coordination with the Acquisition, Project Management, Support Systems, and Software Engineering Institute tasks is also required.

2.1.3.5 Deliverables. Deliverables include acquisition and incentive guidelines for using the metrics and models in system acquisition educational programs on the use of the models, metrics and baselines; newsletters describing information available; and status reports which summarize report distribution activities, tool availability and usage, and results of training sessions and clinics.

2.1.3.6 Cost Factors. Leverage should be achieved by using non-STARS Programs to provide much of the data collection and analysis needed by this task area, once the basic value of measurement is demonstrated.

2.1.3.7 Benefit. Benefits include the insertion of measurement technology as an integral part of the acquisition process and the dissemination of measurement related information, thereby stimulating the improvement of software engineering practices on mission critical systems.

2.1.4 Support Use of Models and Metrics by STARS

2.1.4.1 Purpose/Goals/Rationale. The measurement task plan supports the use of metrics throughout the STARS Program. This support focuses the interaction of the measurement task area with the other task areas. This focus assures that the measurement needs of the whole STARS program and the other task areas are being met and expedites measurement related technology insertion between task areas.

2.1.4.2 Input. The measurement task area should build upon its existing knowledge base in assisting the other STARS task areas.

STARS candidate tasks should be submitted for evaluation, and input required from the other STARS task areas will be a definition of their measurement needs and feedback on the measurement support being provided.

2.1.4.3 Description.

Determine Return on DoD Investment

Quantitative assessments of STARS candidate projects would be made to help determine their inclusion in the Program. Periodically, quantitative evaluations of the progress due to STARS technology and methodologies will also be made.

Define Areas of Needed Support.

Links with the activities sponsored by the other task areas would be established as a part of the STARS program and the areas of needed support will be identified and coordinated. Customized measurement and analysis activities will be sponsored to provide a quantitative analysis of STARS prototypes.

2.1.4.4 Coordination. Since this sub-STARS task requires input from the other task areas, coordination on an as yet unspecified basis is necessary.

2.1.4.5 Benefit. The major benefits of this subtask is that it might assure that the STARS Program will contain high payoff efforts whose value can be defended, and proven within a relatively short period of time.

3.0 OPPORTUNITIES

3.1 Upcoming Conferences/Workshops

IEEE Computer Society Workshop on Software Engineering Technology Transfer, April 25-27,1983, Konover Hotel, Miami Beach Florida.

3.2 Information Resources

[1] Wasserman, A.I., ed. Special Issue on Automated Development Environments. in Computer, Vol.14,no.4, April 1981, pp.7-54

[2] Buxton and Stenning. Requirements for Ada Programming Support Environments

[3] Deutsch and Taft, eds. Requirements for an Experimental Programming Environment

[4] McCall and Matsumoto. Software Quality. Vols.1 and 2, Metrics Enhancement, and Measurement Manual

[5] Pingree Park Conference - Journal of System Software and January 82 ACM Software Engineering Notes

[6] References noted in Measurement Appendix

[7] Brooks, R.E. Studying Programmer Behavior Experimentally: The Problems of Proper Methodology, Comm. of the ACM, Vol.23,No.4, April 1980, pp.207-213.

[8] Sheil, B.A. The Psychological Study of Programming, ACM Computing Surveys, Vol.13, No.1, March 1981, pp.101-120.

[9] Reports describing the work of the IEEE Computer Society working group on software reliability standards

[10] A second Ada Letters which describes preliminary Ada specific metric project results DACS newsletters

[12] Software Development Methodologies and Ada - AJPO - Nov.82 Report by Freeman and Wasserman

[13] DACS reports

[14]NASA-SEL reports

3.3 Current DoD and non-DoD Activities

Human Resources - the results of the Navy Material Command Workshop which addressed skill level needs (October 1982) and the RFP based on that workshop.

Project Management - reports from the AJPO initiated work on the identification of the initial tool set to be introduced using the NBS tool taxonomy and the results of Project Management planning.

END

DATE
FILMED

6-83

DTIC